



日本国特許庁  
JAPAN PATENT OFFICE

#4

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office

出願年月日

Date of Application:

2001年 7月25日

出願番号

Application Number:

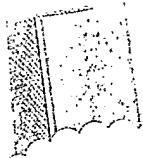
特願2001-224250

出願人

Applicant(s):

株式会社ソニー・コンピュータエンタテインメント

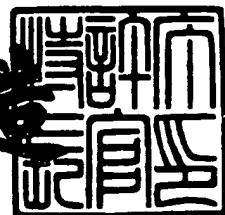
CERTIFIED COPY OF  
PRIORITY DOCUMENT



2001年 8月31日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕



**【書類名】** 特許願  
**【整理番号】** SCEI01063  
**【提出日】** 平成13年 7月25日  
**【あて先】** 特許庁長官殿  
**【国際特許分類】** G06F 11/00  
                        G06F 9/06  
**【発明の名称】** 情報処理システム、実行可能モジュール生成方法および記憶媒体  
**【請求項の数】** 24  
**【発明者】**  
    **【住所又は居所】** 東京都中野区中央一丁目38番1号 住友中野坂上ビル  
                        株式会社シュガーアンドロケッツ内  
    **【氏名】** 山本 浩  
**【発明者】**  
    **【住所又は居所】** 東京都中野区中央一丁目38番1号 住友中野坂上ビル  
                        株式会社シュガーアンドロケッツ内  
    **【氏名】** 大平 俊充  
**【特許出願人】**  
    **【識別番号】** 395015319  
    **【氏名又は名称】** 株式会社 ソニー・コンピュータエンタテインメント  
**【代理人】**  
    **【識別番号】** 100084032  
    **【弁理士】**  
    **【氏名又は名称】** 三品 岩男  
    **【電話番号】** 045(316)3711  
**【選任した代理人】**  
    **【識別番号】** 100087170  
    **【弁理士】**  
    **【氏名又は名称】** 富田 和子

【電話番号】 045(316)3711

【先の出願に基づく優先権主張】

【出願番号】 特願2000-246404

【出願日】 平成12年 8月15日

【手数料の表示】

【予納台帳番号】 011992

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9912211

【ブルーフの要否】 要

【書類名】明細書

【発明の名称】情報処理システム、実行可能モジュール生成方法および記憶媒体

【特許請求の範囲】

【請求項1】

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含む、暗号化済みのプロテクト対象オブジェクトを記憶した記憶手段と、

前記記憶手段から前記暗号化済みプロテクト対象オブジェクトを読み出し、当該暗号化済みプロテクト対象オブジェクトを復号する復号手段と、

前記復号後のプロテクト対象オブジェクトと他のオブジェクトとの結合によって作成される実行可能モジュールに前記プロテクトコードを含ませるコード書込み手段と、

前記復号後のプロテクト対象オブジェクトを、前記他のオブジェクトとの結合を行った後に消去する削除手段と、

を有することを特徴とする情報処理装置。

【請求項2】

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含む、暗号化済みのプロテクト対象オブジェクトを記憶した記憶手段と、

前記記憶手段から前記暗号化済みプロテクト対象オブジェクトを読み出し、当該暗号化済みプロテクト対象オブジェクトを復号する復号手段と、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成するコード生成手段と、

前記復号後のプロテクト対象オブジェクトに前記第一のプロテクトコードを埋め込み、当該第一のプロテクトコード埋込み済みのプログラム対象オブジェクトと他のオブジェクトとの結合により実行可能モジュールが作成されたら、当該実行可能モジュールに、前記第二のプロテクトコードを埋め込むコード書込み手段と、

前記第一のプロテクトコード埋込み済みのプロテクト対象オブジェクトを、前

記二のプロテクトコード埋込み前に消去する削除手段とを有することを特徴とする情報処理装置。

【請求項3】

請求項2記載の情報処理装置であって、  
前記コード生成手段は、  
乱数から、前記第一のプロテクトコードと前記第二のプロテクトコードを生成  
することを特徴とする情報処理装置。

【請求項4】

請求項2または3記載の情報処理装置であって、  
前記コード書き込み手段は、  
前記第一のプロテクトコードと前記第二のプロテクトコードとにダミーデータ  
を付加することを特徴とする情報処理装置。

【請求項5】

請求項1、2、3および4のいずれか1項に記載の情報処理装置であって、  
前記コード書き込み手段は、前記実行可能モジュールに含ませるためのプロテク  
トコードを暗号化し、  
前記プロテクト対象オブジェクトには、前記実行可能モジュールを読み込んだ  
装置に、当該実行可能モジュールに含まれた暗号化済みのプロテクトコードを、  
当該プロテクトコードの診断に際して復号させるための手続きを含む、  
ことを特徴とする情報処理装置。

【請求項6】

情報処理装置に処理を実行させるためのプログラムが格納された、機械読み取  
り可能な記憶媒体であって、  
前記プログラムは、前記情報処理装置に、  
実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させ  
る手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オ  
ブジェクトの復号により生成する復号処理と、  
前記復号処理で生成されたプロテクト対象オブジェクトと他のオブジェクトと  
の結合により、前記実行可能モジュールを作成する結合処理と、

前記結合処理による作成される実行可能モジュールに前記プロジェクトコードを含ませるためのコード書込み処理と、

前記復号処理で生成されたプロジェクト対象オブジェクトを、前記他のオブジェクトとの結合後に消去する削除処理と、  
を実行させることを特徴とする記憶媒体。

【請求項7】

情報処理装置に処理を実行させるためのプログラムが格納された、機械読み取り可能な記憶媒体であって、

前記プログラムは、前記情報処理装置に、  
実行可能モジュールに含まれた複数のプロジェクトコードの矛盾により処理を終了させる手続きを含むプロジェクト対象オブジェクトを、暗号化済みのプロジェクト対象オブジェクトの復号により生成する復号処理と、

互いに関連付けられた第一のプロジェクトコードと第二のプロジェクトコードとを生成するコード生成処理と、

前記復号処理後、当該復号処理で生成されたプロジェクト対象オブジェクトに、前記第一のプロジェクトコードを埋め込む第一コード書込み処理と、

前記コード書込み処理後、当該コード書込み処理で前記第一のプロジェクトコードが埋め込まれたプロジェクト対象オブジェクトと、他のオブジェクトとの結合により、実行可能モジュールを作成する結合処理と、

前記結合処理後、当該結合処理で作成された前記実行可能モジュールに前記第二プロジェクトコードを埋め込む第二コード書込み処理と、

前記第一コード書込み処理後、前記第二コード書込み処理前に、前記復号処理で生成された前記プロジェクト対象オブジェクトを消去する削除処理と、

を実行させることを特徴とする記憶媒体。

【請求項8】

請求項7記載の記憶媒体であって、

前記プログラムは、前記コード生成処理において、前記情報処理装置に、乱数から、前記第一のプロジェクトコードと前記第二のプロジェクトコードを生成させることを特徴とする記憶媒体。

【請求項9】

請求項7または8記載の記憶媒体であって、  
前記プログラムは、前記情報処理装置に、  
前記第一のプロテクトコードと前記第二のプロテクトコードとにダミーデータ  
を付加させることを特徴とする記憶媒体。

【請求項10】

請求項6、7、8および9のいずれか1項に記載の記憶媒体であって、  
前記プログラムは、前記実行可能モジュールに含ませるためのプロテクトコードを暗号化する処理を前記情報処理装置に実行させ、  
前記プロテクト対象オブジェクトには、前記実行可能モジュールを読み込んだ装置に、当該実行可能モジュールに含まれた暗号化プロテクトコードを、当該プロテクトコードの診断に際して復号させるための手続きが含まれることを特徴とする記憶媒体。

【請求項11】

情報処理装置によって処理されるオブジェクトを記憶した、機械読み取り可能な記憶媒体であって、  
暗号化されたプロテクト対象オブジェクトが格納され、  
当該プロテクト対象オブジェクトは、自身を組み込んだ実行可能モジュールに含まれた1以上のプロテクトコードに矛盾があれば処理を終了させる手続きを含む、  
ことを特徴とする記憶媒体。

【請求項12】

請求項11記載の記憶媒体であって、  
前記プロテクト対象オブジェクトは、  
前記実行可能モジュールに含まれているプロテクトコードが暗号化されている場合に、当該プロテクトコードの診断に先立ち、当該プロテクトコードを復号させるための手続きを含む、  
ことを特徴とする記憶媒体。

【請求項13】

複数のオブジェクトを結合して、実行可能モジュールを生成する処理を情報処理装置に実行させる実行可能モジュール生成方法であって、

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成し、

前記復号後のプロテクト対象オブジェクトと他のオブジェクトとの結合と、前記プロテクトコードの書き込みとによって、前記実行可能モジュールを生成し、

前記復号後のプロテクト対象オブジェクトは、前記他のオブジェクトとの結合がなされた後に消去される、

ことを特徴とする実行モジュール生成方法。

【請求項14】

複数のオブジェクトを結合して、実行可能モジュールを生成する処理を情報処理装置に実行させる実行可能モジュール生成方法であって、

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成し、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成して、前記復号後のプロテクト対象オブジェクトに当該第一のプロテクトコードを埋め込み、当該第一のプロテクトコード埋込み済みのプログラム対象オブジェクトと他のオブジェクトとの結合により実行可能モジュールを作成したら、当該実行可能モジュールに当該第二のプロテクトコードを埋め込み、

前記第一のプロテクトコード埋込み済みのプロテクト対象オブジェクトは、前記二のプロテクトコード埋込み前に消去されることを特徴とする実行可能モジュール生成方法。

【請求項15】

複数のオブジェクトの結合により組み立てられた実行可能モジュールを実行可能な装置が実行するための実行可能モジュールが格納された、機械読み取り可能な記憶媒体であって、

前記複数のオブジェクトのうちの、少なくとも1のオブジェクトには、

少なくとも1つのプロジェクトコードに矛盾があるか否かを診断し、当該診断結果に応じて、前記実行可能モジュールの処理を終了させる手続きを含んだライブリオブジェクトが含まれ、

前記実行可能モジュールには、

少なくとも1つのプロジェクトコードが埋め込まれている、  
ことを特徴とする記憶媒体。

【請求項16】

複数のオブジェクトの結合によって生成された実行可能モジュールを実行する  
エンタテインメント装置であって、

前記少なくとも1つのオブジェクトに含まれたプロジェクトコードと、それ以外  
のプロジェクトコードとが前記実行可能モジュールに含まれている場合、それら  
のプロジェクトコード間の関係を診断し、当該関係に矛盾があれば、前記実行可能  
モジュールの処理を終了させる手段を有することを特徴とするエンタテインメン  
ト装置。

【請求項17】

情報処理装置に実行されるプログラムであって、

前記情報処理装置に、

実行可能モジュールに含まれたプロジェクトコードの矛盾により処理を終了させ  
る手続きを含むプロジェクト対象オブジェクトを、暗号化済みのプロジェクト対象オ  
ブジェクトの復号により生成する復号処理と、

前記復号処理で生成されたプロジェクト対象オブジェクトと他のオブジェクトと  
の結合により、前記実行可能モジュールを作成する結合処理と、

前記結合処理による作成される実行可能モジュールに前記プロジェクトコードを  
含ませるためのコード書き込み処理と、

前記復号処理で生成されたプロジェクト対象オブジェクトを、前記他のオブジエ  
クトとの結合後に消去する削除処理と、

を実行させることを特徴とするプログラム。

【請求項18】

情報処理装置に実行されるプログラムであって、

前記情報処理装置に、

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成する復号処理と、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成するコード生成処理と、

前記復号処理後、当該復号処理で生成されたプロテクト対象オブジェクトに、前記第一のプロテクトコードを埋め込む第一コード書込み処理と、

前記コード書込み処理後、当該コード書込み処理で前記第一のプロテクトコードが埋め込まれたプロテクト対象オブジェクトと、他のオブジェクトとの結合により、実行可能モジュールを作成する結合処理と、

前記結合処理後、当該結合処理で作成された前記実行可能モジュールに前記第二プロテクトコードを埋め込む第二コード書込み処理と、

前記第一コード書込み処理後、前記第二コード書込み処理前に、前記復号処理で生成された前記プロテクト対象オブジェクトを消去する削除処理と、

を実行させることを特徴とするプログラム。

【請求項19】

請求項18記載のプログラムであって、

前記コード生成処理において、前記情報処理装置に、

乱数から、前記第一のプロテクトコードと前記第二のプロテクトコードを生成させることを特徴とするプログラム。

【請求項20】

請求項18または19記載のプログラムであって、

前記情報処理装置に、

前記第一のプロテクトコードと前記第二のプロテクトコードとにダミーデータを付加させることを特徴とするプログラム。

【請求項21】

請求項17、18、19および20のいずれか1項に記載のプログラムであつて、

前記情報処理装置に、前記実行可能モジュールに含ませるためのプロジェクトコードを暗号化する処理を実行させ、

前記プロジェクト対象オブジェクトには、前記実行可能モジュールを読み込んだ装置に、当該実行可能モジュールに含まれた暗号化プロジェクトコードを、当該プロジェクトコードの診断に際して復号させるための手続きが含まれる、

ことを特徴とするプログラム。

【請求項22】

情報処理装置によって処理されるオブジェクトであって、

暗号化され、かつ、

自身を組み込んだ実行可能モジュールを読み込んだ装置に、当該実行可能モジュールに含まれた1以上のプロジェクトコードに矛盾があれば処理を終了させるための手続きを含む、

ことを特徴とするオブジェクト。

【請求項23】

請求項22記載のオブジェクトであって、

前記実行可能モジュールに含まれているプロジェクトコードが暗号化されている場合に、当該プロジェクトコードに矛盾があるか否かの診断に先立ち、当該プロジェクトコードを前記装置に復号させるための手続きを含む、ことを特徴とするオブジェクト。

【請求項24】

複数のオブジェクトの結合により組み立てられた実行可能モジュールを実行可能な装置が実行する実行可能モジュールであって、

少なくとも1つのプロジェクトコードが埋め込まれ、

前記複数のオブジェクトには、

前記装置に、前記実行可能モジュールに含まれているプロジェクトコードに矛盾があるか否かを診断させ、当該診断結果に応じて、前記実行可能モジュールの処理を終了させる手続きを含んだライブラリオブジェクトが含まれる、

ことを特徴とする実行可能オブジェクト。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、プログラム開発環境等におけるリソース不正コピーの防止技術に関する。

【0002】

【従来の技術】

プログラムの不正コピーを防止するための技術として、特許公報第2569564号記載の、ソフトウェアのコピープロテクト装置が知られている。この装置によれば、フロッピディスクから読み出したプロテクトコードが外部に出力されないため、その不正な書替えを防止することができる。これにより、不正に複製されたソフトウェアを実行可能とすることが困難となり、ソフトウェアの不正な複製を効果的に防止することができる。

【0003】

【発明が解決しようとする課題】

プログラムが作成される際には、通常、グラフィック機能その他の基本的機能を提供するライブラリ等のリソースが使用される。このような、ライブラリ等のリソースに含まれているオブジェクトは、実行可能モジュールではない。そのため、不正コピー防止のためのプロテクトをライブラリにかけることが困難である。したがって、現在のところ、ライブラリ等のリソースは、メーカー側がプログラム開発者の良識を信頼するというかたちで、プロテクトがかけられていないまま提供されているのが実情である。

【0004】

しかし、ライブラリ等のリソースは、プログラムと同様、重要な開発成果であるため、なんらかのコピープロテクトがかけられるようにすることが望まれている。

【0005】

そこで、本発明は、実行可能モジュール以外のオブジェクトの不正コピーを防止することを目的とする。

【0006】

【課題を解決するための手段】

上記課題を解決するため、本発明は、

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含む、暗号化済みのプロテクト対象オブジェクトを記憶した記憶手段と、

前記記憶手段から前記暗号化済みプロテクト対象オブジェクトを読み出し、当該暗号化済みプロテクト対象オブジェクトを復号する復号手段と、

前記復号後のプロテクト対象オブジェクトと他のオブジェクトとの結合によって作成される実行可能モジュールに前記プロテクトコードを含ませるコード書込み手段と、

前記復号後のプロテクト対象オブジェクトを、前記他のオブジェクトとの結合を行った後に消去する削除手段と、

を有することを特徴とする情報処理装置を提供する。

【0007】

【発明の実施の形態】

以下、添付の図面を参照しながら、本発明に係る実施の一形態について説明する。ただし、ここでは、ライブラリをプロテクト対象オブジェクトとする。

【0008】

まず、図1により、プログラム開発側で使用される情報処理システムの概略ハードウェア構成について説明する。

【0009】

プログラム開発側の情報処理システムは、情報処理装置100、ユーザ(プログラム開発者)からのデータ入力を受け付ける入力装置(マウス、キーボード等)120、各種データを出力する出力装置(ディスプレイ装置等)130、等から構成されている。さらに、このシステムには、その他の装置が必要に応じて増設されることもある。

【0010】

情報処理装置100は、後述の実行可能プログラムファイル作成処理等を実行するCPU101、グラフィック描画処理等を実行するグラフィックス機構10

2、主記憶103、CPU101と主記憶103との間をつないだプロセッサバス105、CPU101等と主記憶103またはバス106との間のデータ転送を制御するメモリ／バス制御チップ104、グラフィックス機構102のグラフィックス制御チップとメモリ／バス制御チップ104との間をつないだAGP、ソフトウェア開発のために必要となる各種データ(シリアル番号等の識別情報が埋め込まれたフロントエンドツール、フロントエンドツールによって起動されるリンクおよびコンパイラ、フロントエンドツールとともに提供された暗号化ライブラリファイル等)が格納されたハードディスク106、ハードディスク106に対するデータ読み込み／書き込み処理を実行するハードディスク・ドライブ107、ハードディスクドライブ107がつながれた高速バス106、記憶媒体Aに対するデータ読み込み書き込み処理を実行するドライブ112、入力装置120からのデータ転送を制御するコントローラ111、出力装置130へのデータ転送を制御するコントローラ113、各コントローラ111,113とドライブ112とがつながれた低速バス109、低速バス109と高速バス106との間をつないだブリッジ回路110、通信ポート(シリアルポート、パラレルポート等)、等を備えている。なお、ここで示した情報処理装置100の内部構成は、ソフトウェア開発に使用されるものとしての一例である。

#### 【0011】

つぎに、この情報処理システムのハードディスク106内の格納データについて説明する。

#### 【0012】

コンパイラは、CPUに、指定ソースプログラムファイルに記述されたソースを、解析処理(字句解析、構文解析、意味解析)等によってオブジェクトとして翻訳編集させ、そのオブジェクトをオブジェクトファイルに格納させる処理が定義されたソフトウェアである。リンクは、CPUに、指定オブジェクトファイル群のそれぞれに記述されているオブジェクトを結合編集させることによって実行可能モジュールを組み立てさせ、その実行可能モジュールを実行可能プログラムファイルに格納させる処理が定義されたソフトウェアである。そして、本実施の形態に係るフロントエンドツールは、CPUに、コンパイラおよびリンクを起動さ

せ、後述の実行可能プログラムファイル作成処理を実行させる処理が定義されたソフトウェアである。

#### 【0013】

暗号化ライブラリファイル $L_1$ （以下、暗号化 $L_i b$ ファイル $L_1$ と呼ぶ）は、フロントエンドツールとともに、ユーザ（プログラム開発者）に提供される。例えば、暗号化 $L_i b$ ファイル $L_1$ が記録されたインストール用CD-ROMと、フロントエンドツールが記録されたインストール用フロッピィディスクとがワンセットとしてユーザに提供される。

#### 【0014】

この暗号化 $L_i b$ ファイル $L_1$ は、1以上のライブラリファイル $L_2$ （以下、 $L_i b$ ファイル $L_2$ と呼ぶ）を適当な暗号化処理で暗号化することによって作成されたものである。ユーザは、実行可能プログラムファイル作成時に、この暗号化 $L_i b$ ファイル $L_1$ をフロントエンドツールで復号化することによって、実行可能プログラムファイル作成に必要な1以上の $L_i b$ ファイル $L_2$ を得る。このように、ライブラリファイルを、暗号化された状態で提供し、それと共に提供されたフロントエンドツールによってその復号化させるようにしたのは、後述するように、フロントエンドツールを持たないユーザによる $L_i b$ ファイル不正使用を禁じるためである。

#### 【0015】

暗号化 $L_i b$ ファイル $L_1$ の復号により生成される $L_i b$ ファイル $L_2$ には、グラフィック機能、イベント処理等、アプリケーションにおいて使用される基本的な関数等のオブジェクトが部品化されて格納されている。また、すべてのアプリケーションにおいて呼び出される初期化関数が含まれている。 $L_i b$ ファイル $L_2$ 内のオブジェクトには、さらに、初期化関数において呼び出されるプロテクトコードチェック手続きと、オブジェクトファイル $M_2$ との結合編集前に第一プロテクトコード $C_1$ が格納される第一プロテクトコード格納変数とが定義されている。ここで、プロテクトコードチェック手続きには、第一プロテクトコード格納変数の格納データとプログラムファイル末尾の格納データとが、正規のプロテクトコード $C_1, C_2$ が満たす関係を満たしているか否かをチェックし、その関係を

満たしていなければ実行中プログラムを初期化段階で終了させる手続きが定義されている。

## 【0016】

なお、本実施の形態では、初期化関数が含まれている L i b ファイル  $L_2$  だけにプロテクトコードチェック関数および第一プロテクトコード格納変数を含ませているが、すべての L i b ファイルにプロテクトコードチェック関数および第一プロテクト格納変数を含ませることにしてもよい。または、代表的な L i b ファイル、例えば、プロテクトをかける必要のある L i b ファイルだけにプロテクトコードチェック関数および第一プロテクト格納変数を含ませることにしてもよい。

## 【0017】

本実施の形態に係る情報処理システムは、このようなハードウェア構成およびハードディスク格納データ(フロントエンドツール、コンパイラ、リンク等のソフトウェア、暗号化ライブラリファイル)によって、図2に示すような機能構成を実現する。具体的には、(1)フロントエンドツールの起動コマンドの入力を受け付ける入力受付部 200、(2)起動時にフロントエンドツール複製チェック処理を実行する起動時チェック処理部 201、(3)開発成果(ソースプログラムファイル  $M_1$ 、オブジェクトファイル  $M_2$ 、実行可能プログラムファイル  $M_3$ )を格納するための開発成果記憶部 202、(4)ソースプログラムファイル  $M_1$  からオブジェクトファイル  $M_2$  を生成するコンパイル処理部 203、(5)暗号化 L i b ファイル  $L_1$  を記憶したライブラリ記憶部 204、(6)暗号化 L i b ファイル  $L_1$  を複号する復号化処理部 205、(7)復号化処理部 204 の復号化処理によって生成された L i b ファイルを一時ファイル  $L_2$  として記憶する一時ファイル記憶部( tmp ) 206、(8) L i b ファイル  $L_2$  とオブジェクトファイル  $M_2$  との結合編集により実行可能プログラムファイル  $M_3$  を生成するリンク処理部 207、(9)リンク処理部 207 の結合編集処理後に一時ファイル  $L_2$  を削除する一時ファイル削除処理部 208、(10)リンク処理部 207 の結合編集処理前の L i b ファイル  $L_2$  とこの L i b ファイル  $L_2$  から作成された実行可能プログラムファイル  $M_3$  とにそれぞれプロテクトコード  $C_1, C_2$  を付与するコード付与処理部 209、(

11)これら各処理部を制御する制御処理部210、が実現される。

#### 【0018】

ここで、コード付与処理部209は、(12)互いに関連付けられた1組のプロテクトコード $C_1, C_2$ (例えば、 $C_1 = f(C_2)$ の関係を満たす1組のプロテクトコード $C_1, C_2$ )を生成するコード生成処理部209a、(13)コード生成処理部209aが生成した1組のプロテクトコード $C_1, C_2$ のうちの一方のコード $C_1$ (第一プロテクトコードと呼ぶ)をL1bファイル $L_2$ 内のコード格納変数に書き込むとともに他方のコード $C_2$ (第二プロテクトコードと呼ぶ)を実行可能プログラムファイル $M_3$ の末尾に付加するコード書込処理部209b、からなっている。

#### 【0019】

つぎに、これら各機能構成部により実行される実行可能プログラムファイル作成処理について説明する。

#### 【0020】

プログラム開発者は、コーディングが終了したら、ソースプログラムファイル $M_1$ をハードディスク108の所定の領域(図2の開発成果記憶部202に相当)に格納してから、そのソースプログラムファイル名と必要なライブラリファイル名とを指定してフロントエンドツールを起動する。これにより、CPUが、フロントエンドツールに定義された処理を実行し、以下に示す実行可能プログラムファイル作成処理(図3のフローチャート参照)が実現される。

#### 【0021】

まず、フロントエンドツールの起動コマンドを受け付けた入力受付部200は、制御処理部210に処理開始を指示する(S300)。制御処理部210は、この指示を受け付けると、フロントエンドツール複製チェック処理の実行を起動時チェック処理部201に指示する。起動時チェック処理部201は、この指示にしたがって、以下に示す方法により、フロントエンドツールの不正使用をチェックする(S301)。

#### 【0022】

例えば、プログラム開発環境にLANが構築されており、ネットワーク上で起動中のプログラムの識別情報が重複していないか否かを監視するチェックプログ

ラムがネットワーク上に常駐している場合であれば、起動時チェック処理部201は、フロントエンドツールに付与されている識別情報をネットワーク上に送信する。これに応じて、チェックプログラムは、それと同じ識別情報がネットワーク上に存在するか否かを返信する。フロントエンドツールがコピー品であれば、同じ識別情報が複数ネットワーク上に存在し、フロントエンドツールがコピー品でなければ、同じ識別情報が複数ネットワーク上に存在することはないため、フロントエンドツールの不正使用のチェックが可能となる。

#### 【0023】

また、フロントエンドツールにdongleが割り当てられている場合であれば、起動時チェック処理部201は、情報処理装置の通信ポートに装着されているdongleに認証メッセージを送る。このメッセージに応じて、dongleは、自身に割り当てられた識別情報を返信する。認証されたユーザがフロントエンドツールを起動した場合であれば、dongleから返信された識別情報が、フロントエンドツールの識別情報と一致し、認証されたユーザ以外のユーザがフロントエンドツールを起動した場合であれば、dongleの装着がないために識別情報の返信がないか、または、dongleから返信された識別情報がフロントエンドツールの識別情報と一致しないことになるため、フロントエンドツールの不正使用のチェックが可能となる。

#### 【0024】

このようなチェックの結果に基づき、起動時チェック処理部201は、フロントエンドツールの使用に不正(フロントエンドツールが複製品である、認証ユーザ以外のユーザによる使用である、等)があるか否かを判断する(S302)。

#### 【0025】

その結果、起動時チェック処理部201がフロントエンドツールの使用に不正があると判断した場合には、制御処理部210が、エラーメッセージを出力してから(S303)、処理を終了させる(S308)。これにより、フロントエンドツールの不正使用が禁止されるため、フロントエンドツールの不正使用者には、暗号化L<sub>i</sub>bファイルL<sub>1</sub>を復号することが困難となる。したがって、L<sub>i</sub>bファイルL<sub>2</sub>の不正使用も禁止される。

## 【0026】

一方、起動時チェック処理部201がフロントエンドツールの使用に不正がないと判断した場合には、制御処理部210が、指定のソースプログラムファイルM<sub>1</sub>の機械語翻訳をコンパイル処理部203に指示する。コンパイル処理部203は、この指示に応じて、指定のソースプログラムファイルM<sub>1</sub>をアセンブリプログラムにコンパイルし、そのアセンブリプログラムをオブジェクトにアセンブルする。その後、コンパイル処理部203は、このとき生成したオブジェクトファイルM<sub>2</sub>を開発成果記憶部202に格納してから、制御処理部210に正常終了を報告する。

## 【0027】

制御処理部210は、コンパイル処理部203から正常終了報告を受け付けると、暗号化L<sub>i b</sub>ファイルL<sub>1</sub>の復号化を復号化処理部205に指示する。復号化処理部205は、この指示に応じて、ライブラリ記憶部204から暗号化L<sub>i b</sub>ファイルL<sub>1</sub>を取り出し、それを復号化する。これにより1以上のL<sub>i b</sub>ファイルL<sub>2</sub>が生成されると、復号化処理部205は、そのL<sub>i b</sub>ファイルL<sub>2</sub>を、適当なファイル名の一時ファイルとして一時ファイル記憶部206に格納し、制御処理部210に正常終了を報告する。

## 【0028】

制御処理部210は、復号化処理部205からの正常終了報告を受け付けると、L<sub>i b</sub>ファイルL<sub>2</sub>に対するプロテクトコード付与をコード付与処理部209に指示する。この指示に応じて、コード付与処理部209では、以下の処理が実行される。まず、コード生成処理部209aが、図4に示すように、乱数Cを発生し、この乱数Cを2つの加工方法F<sub>1</sub>, F<sub>2</sub>(例えば、乱数Cを入力データとする2つの関数等)でそれぞれ加工することによって2つのプロテクトコードC<sub>1</sub>, C<sub>2</sub>を生成する。ここで生成された2つのプロテクトコードC<sub>1</sub>, C<sub>2</sub>には、前述したように、予め定められた関係(ここでは、C<sub>1</sub> = f(C<sub>2</sub>)とする)がある。これらのプロテクトコードC<sub>1</sub>, C<sub>2</sub>は、さらに、コード生成処理部209aによって、適当な暗号化方法でそれぞれ暗号化される。ついで、コード書き込み処理部209bが、第二プロテクトコードC<sub>2</sub>を保持するとともに、所定のL<sub>i b</sub>ファイルL<sub>2</sub>内

の第一プロテクトコード格納変数に第一プロテクトコードC<sub>1</sub>を格納する(S304)。

#### 【0029】

このようにして、LibファイルL<sub>2</sub>に対するプロテクトコード付与が終了すると、制御処理部210は、リンク処理部207に、オブジェクトファイルM<sub>2</sub>と指定のLibファイルL<sub>2</sub>との結合編集を指示する。リンク処理部207は、この指示に応じて、オブジェクトファイルM<sub>2</sub>と指定のLibファイルL<sub>2</sub>とを結合し、実際の番地の埋込み等を行う。これにより実行可能プログラムファイルM<sub>3</sub>が組み立てられると、リンク処理部207は、その実行可能プログラムファイルM<sub>3</sub>を、適当な名前で開発成果記憶部202に格納してから、正常終了を制御処理部210に報告する(S305)。

#### 【0030】

制御処理部210は、リンク処理部207からの正常終了報告を受け付けると、一時ファイル削除を一時ファイル削除処理部208に指示する。この指示に応じて、一時ファイル削除処理部208は、すべてのLibファイルL<sub>2</sub>を一時ファイル記憶部206から削除する(S306)。

#### 【0031】

その後、制御処理部210は、実行可能プログラムファイルL<sub>3</sub>に対するプロテクトコード付与をコード付与処理部209に指示する。この指示に応じて、コード付与処理部209のコード書き込み処理部209bが、保持中の第二プロテクトコードC<sub>2</sub>を実行可能プログラムファイルM<sub>3</sub>に付加してから、正常終了を制御処理部210に報告する(S307)。なお、ライセンシー管理(例えば、プログラム作成会社の識別等)のために、このときフロントエンドツールの識別情報も、コード付与処理部209によって実行可能プログラムファイルM<sub>3</sub>に埋め込まれるようにしてもよい。

#### 【0032】

制御処理部210は、コード付与処理部209からの正常終了報告を受け付けると、処理を終了させる(S308)。なお、以上の処理により作成された実行可能プログラムファイルM<sub>3</sub>は、例えば、光ディスク等の記憶媒体に焼き付けられ

て市場に供給される(図9参照)。

### 【0033】

本実施の形態に係る実行可能プログラムファイル作成処理によれば、以下に示すように、L i bファイルL<sub>2</sub>の不正使用が防止される。

(1) ハードディスクに格納されたL i bファイルが暗号化されており、かつ、それを復号するフロントエンドツールの不正使用が、起動時チェック(図2のS301)によって禁止される。このため、フロントエンドツールの正当使用者以外のユーザによる、L i bファイルの使用が防止される。

(2) 図5に示すように、実行中のプログラムファイルに埋め込まれたプロテクトコードに矛盾があった場合にその処理を終了させるプロテクトコードチェック手続きEをL i bファイルL<sub>2</sub>にあらかじめ含めておき、フロントエンドツールの使用によってL i bファイルL<sub>2</sub>とオブジェクトファイルM<sub>2</sub>とが結合された実行可能プログラムファイルM<sub>3</sub>にだけ、実行可能プログラムファイル作成処理の過程(S304、S307)において適正なプロテクトコードC<sub>1</sub>、C<sub>2</sub>が埋め込まれるようにすることによって、フロントエンドツールの正当使用により作成された実行可能プログラムファイルM<sub>3</sub>のみを実行可能とすることができます。すなわち、フロントエンドツールによらずL i bファイルが結合された実行可能プログラムファイル、具体的には、不正なプロテクトコード(一方または両方)が埋め込まれた実行可能プログラムファイル、プロテクトコード(一方または両方)が埋め込まれていない実行可能プログラムファイル等の実行が、初期化段階において阻止される。このため、フロントエンドツールの正当使用者以外のユーザによる、L i bファイルの使用が防止される。

(3) 図5に示すように、乱数Cから生成された1組のプロテクトコードC<sub>1</sub>、C<sub>2</sub>のうち、第一プロテクトコードC<sub>1</sub>だけをL i bファイルL<sub>2</sub>に埋め込んでおき(S304)、その後、L i bファイルL<sub>2</sub>とオブジェクトファイルM<sub>2</sub>との結合編集(S305)が終了したら、実行可能プログラムファイルM<sub>2</sub>へ第二プロテクトコードC<sub>2</sub>を埋め込む前にL i bファイルL<sub>2</sub>を消去(S306)してしまうため、使用済みL i bファイルL<sub>2</sub>の不正利用が防止される。すなわち、一方のプロテクトコードC<sub>1</sub>が埋め込まれた使用済みL i bファイルL<sub>2</sub>を消去前に取り出すこ

とができたとしても、その段階(S304～S306)では、乱数Cから生成された他方のプロテクトコードC<sub>2</sub>の取り出しが不可能であるから、その使用済みL<sub>i b</sub>ファイルL<sub>2</sub>に埋め込まれている第一プロテクトコードC<sub>1</sub>と矛盾しない第二プロテクトコードC<sub>2</sub>を知ることは困難である。このため、消去前に取り出した使用済みL<sub>i b</sub>ファイルL<sub>2</sub>とオブジェクトファイルとを結合編集したとしても、使用済みL<sub>i b</sub>ファイルL<sub>2</sub>に埋込済みの第一プロテクトコードと適正な関係を有する第二プロテクトコードを実行可能プログラムファイルに埋め込むことは困難である。したがって、使用済みL<sub>i b</sub>ファイルの不正使用が防止される。

(4)最終的に作成された実行可能プログラムファイルM<sub>3</sub>の解析によって、それに埋め込まれた1組のプロテクトコードC<sub>1</sub>,C<sub>2</sub>を知ってから、つぎのフロントエンドツール起動時に使用済みL<sub>i b</sub>ファイルL<sub>2</sub>を消去前に取り出したとしても、プロテクトコードC<sub>1</sub>,C<sub>2</sub>は、毎回、乱数から生成されるため、消去前に取り出した使用済みL<sub>i b</sub>ファイルL<sub>2</sub>に埋め込まれている第一プロテクトコードC<sub>1</sub>に対応する第二プロテクトコードは、実行可能プログラムファイルM<sub>3</sub>の解析によって知った第二プロテクトコードC<sub>2</sub>とは必ずしも一致しない。このため、最終的に作成された実行可能プログラムファイルM<sub>3</sub>の解析によっても、適正な関係を有する1組のプロテクトコードを得ることは困難である。したがって、消去前に取り出した使用済みL<sub>i b</sub>ファイルL<sub>2</sub>とオブジェクトファイルとを結合編集したとしても、使用済みL<sub>i b</sub>ファイルL<sub>2</sub>に埋込済みの第一プロテクトコードと適正な関係を有する第二プロテクトコードを実行可能プログラムファイルに埋め込むことは困難である。このため、使用済みL<sub>i b</sub>ファイルの不正使用が防止される。

#### 【0034】

以上の4つの防止策によってL<sub>i b</sub>ファイル不正使用が防止されるが、ライブラリバージョンアップのたびに、L<sub>i b</sub>ファイルの暗号化方法とプロテクトコードチェック手続きの内容とをそれぞれ変更するようにすれば、L<sub>i b</sub>ファイルの不正使用防止が強化される。なお、このようにする場合には、L<sub>i b</sub>ファイル複合方法および乱数加工方法が変更されたフロントエンドツールが、新バージョンの暗号化L<sub>i b</sub>ファイルとともにリリースされる必要がある。

## 【0035】

ところで、以上においては、L i b ファイル  $L_2$  内で定義された第一プロテクトコード格納変数に第一プロテクトコード  $C_1$  を格納し、実行可能プログラムファイル  $M_3$  の末尾に第二プロテクトコード  $C_2$  を付加することとしているが、必ずしも、このようにする必要はない。例えば、プロテクトコード付与処理部が、図6に示すように、L i b ファイル  $L_2$  の末尾に、ダミーデータと第一プロテクトコード  $C_1$  とを含んだ第一ダミーエリア①を付加し、実行可能プログラムファイル  $M_3$  の末尾に、ダミーデータと第二プロテクトコード  $C_2$  とを含んだ第二ダミーエリア②を付加するようにしてもよい。そして、2つのダミーエリア①②の合計データ量を変えないように、各ダミーエリア①②に含まれるダミーデータのデータ量をランダムに変化させるようにすれば、1つの実行可能プログラムファイル  $M_3$  を作成するごとに、その内部におけるプロテクトコード  $C_1, C_2$  のアドレスが変化するため、プロテクトコード  $C_1, C_2$  の間にある関係の解析を困難にすることができる。

## 【0036】

また、以上においては、プロテクトコードチェック手続きが組み込まれた L i b ファイルに第一プロテクトコードをリンク前に埋め込み、実行可能プログラムファイルが完成したときに、さらに、この実行可能プログラムファイルに第二プロテクトコードを埋め込むこととしているが、かならずしも、このようにする必要はない。例えば、プロテクトコードチェック手続きが組み込まれた L i b ファイルにはプロテクトコードを組み込まず、実行可能プログラムファイルが完成したときに、この実行可能プログラムファイルに、プロテクトコードチェック手続きによって正当性をチェック可能なプロテクトコード(例えば、互いに関連付けられた複数のプロテクトコード)を埋め込むようにしてもよい。

## 【0037】

つぎに、本実施の形態に係るプログラムファイル作成処理により作成された実行可能プログラムファイル  $M_3$  を実行することができるシステムの概略ハードウェア構成について説明する。ただし、ここでは、システムの一例として、エンタテインメント装置を挙げることとする。

## 【0038】

図7は、本実施の形態に係るエンタテインメント装置700およびその周辺機器の外観図であり、図8は、本実施の形態に係るエンタテインメント装置700の概略ハードウェア構成図である。

## 【0039】

本実施の形態に係るエンタテインメント装置700は、CD-ROM、DVD-ROM等の光ディスク900からゲームプログラム(前述のプログラムファイル作成処理により作成された実行可能プログラムファイルM<sub>3</sub>等)を読み出し、そのゲームプログラムにしたがってゲームを実行するものである。ここで、「ゲームの実行」とは、エンタテインメント装置700が、ユーザ(以下、ゲームプレイヤと呼ぶ)からの指示にしたがって、表示装置(CRT、LCD、プロジェクション装置等)上の表示映像およびオーディオ装置からの音声をゲームストーリ等にあわせて変化させることをいう。

## 【0040】

図7に示すように、エンタテインメント装置700の筐体前面部には、ゲームプレイヤがゲームを実行させるために必要となる各種の操作部が設けられている。このような操作部の具体例としては、ゲームプログラムが格納された光ディスク900が装着されるディスク装着部701、ディスク装着部701からトレイを引き出すためのトレイ操作ボタン701A、ゲームプレイヤからの入力を受け付ける各種入力受付け部(ボタン753A, 753B, 754~758、方向キー752、ステイック751A, 751B等)を有するコントローラ750が接続される複数(ここでは2つ)のコントローラ接続部702A, 702B、実行中のゲームをリセットするためのリセットボタン703、ゲームデータ等を記録させるためのメモリカード730が装着される複数(ここでは2つ)のメモリカード装着部704A, 704B、等が挙げられる。ここでコントローラ接続部702A, 702Bが複数設けられているのは、各コントローラ接続部702A, 702Bにそれぞれコントローラを接続することによって、エンタテインメント装置700が、複数のゲームプレイヤからの指示を受け付けることができるようにするためである。

## 【0041】

また、エンタテインメント装置700の筐体背面部(前面の反対側)には、電源オンオフスイッチ(不図示)、表示装置等が接続されるAV端子(不図示)が設けられている。

## 【0042】

そして、このエンタテインメント装置700の内部には、図8に示すように、装置全体を制御するメインCPU800、AV端子(不図示)から出力されるビデオ信号を生成するグラフィックプロセッサ(GP)801、コントローラ接続部702A,702Bに接続されたコントローラ750およびメモリカード接続部704A,704Bに接続されたメモリカード730とのデータ転送を制御するI/Oプロセッサ(IOP)802、ディスク装着部701に装着された光ディスク900からのデータ読み出しを制御する光ディスク制御部803、AV端子(不図示)から出力されるオーディオ信号を生成するサウンド再生処理プロセッサ(SPU)804、サウンドバッファ805、メインCPU800およびIOP802が実行する各オペレーティングシステムプログラムが格納されたOS-ROM806、IOPのワークエリアとして用いられるIOPメモリ807、メインCPU800のワークエリアおよびバッファとして用いられるメインメモリ808、これら各部の間をつなぎだすバス809等が搭載されている。

## 【0043】

このエンタテインメント装置700に電源が投入されると、2種類のオペレーティングシステムプログラム(メインCPU用、IOP用)がそれぞれOS-ROM806から読みだされ、それらがメインCPU800とIOP802とによってそれぞれ起動される。これにより、オペレーティングシステムによる、装置各部の統合制御が開始され、エンタテインメント装置としての各種機能がゲームプレイヤに提供される。具体的には、光ディスクからの実行可能プログラムファイル読み込みおよびその実行、コントローラを介したゲームプレイヤからの指示受け、ゲームプレイヤの指示に応じた映像表示および効果音・楽音出力等が可能な環境が提供される。

## 【0044】

このような環境において、図9に示した光ディスク900から、上述の実行可能プログラムファイル作成処理によって作成されたゲーム用実行可能プログラムファイルM<sub>3</sub>が読み込まれると、そのゲーム用実行可能プログラムファイルM<sub>3</sub>に定義された処理を実行するための機能構成が、メインCPU800によってプロセスとして実現される。すなわち、図10に示すように、エンタテインメント装置700は、(1)実行可能プログラムファイルM<sub>3</sub>の所定領域(第一プログラム格納変数)の格納データ(実行可能プログラムファイルM<sub>3</sub>がL i bファイルの正当使用により生成されたものであれば、正当な第一プロテクトコードC<sub>1</sub>)が格納された第一プロテクトコード格納部1000、(2)実行可能プログラムファイルM<sub>3</sub>の末尾領域の格納データ(実行可能プログラムファイルM<sub>3</sub>がL i bファイルの正当使用により生成されたものであれば、正当な第二プロテクトコードC<sub>2</sub>)が格納された第二プロテクトコード格納部1001、(3)起動時にプロテクトコードチェック手続きを実行するプロテクトコードチェック処理部1002、(4)プロテクトコードチェック処理部1002からの開始指示に応じて、ゲームを実行するゲーム実行処理部1003、(5)コントローラからの入力信号をゲーム実行処理部1003に渡す入力データ受付処理部1004、(6)ゲーム実行処理部1003の指示にしたがって、効果音等のオーディオ信号を生成する音声制御処理部1005、(7)ゲーム処理部1003の指示にしたがって、ゲーム画面映像のビデオ信号を生成する映像制御処理部1006、が実現される。ここで、プロテクトコードチェック処理部1002には、(8)第一プロテクトコードC<sub>1</sub>および第二プロテクトコードC<sub>2</sub>を復号するコード復号処理部1002a、(9)復号後の2つのプロテクトコードC<sub>1</sub>, C<sub>2</sub>の整合性をチェックするコードマッチング処理部1002b、(10)与えられた終了指示にしたがって、処理を終了させる終了処理部1002d、(11)コードマッチング処理部1002bのチェック結果に応じて、終了処理部1002dに終了指示またはゲーム処理部に開始指示を与える制御処理部1002c、が含まれている。

#### 【0045】

つぎに、図11により、これらの機能構成部によって実行される処理について説明する。

## 【0046】

図9の光ディスク900から読み込まれた実行可能プログラムファイルM<sub>3</sub>が起動されると(S1100)、図10の機能構成部が以下の処理を開始する。

## 【0047】

ゲームの開始前に、実行可能プログラムファイルM<sub>3</sub>に組み込まれたL<sub>i b</sub>ファイルL<sub>2</sub>が正当使用によるものか否かを判断するため、以下の自己診断処理がプロテクトコードチェック処理部1002によって実行される。

## 【0048】

まず、制御処理部1002cが、コードの復号をコード復号処理部1002aに指示する。この指示に応じて、コード復号処理部1002aは、第一プロテクトコード格納部1000の格納データと、第二プログラムコード格納部1001の格納データを取り出し、それらをそれぞれ復号する。

## 【0049】

フロントエンドツールの正当使用により作成された実行可能プログラムファイルM<sub>3</sub>であれば、1組のプロテクトコード生成に用いられた関係(ここではC<sub>1</sub>=F(C<sub>2</sub>))を満たすデータC<sub>1</sub>,C<sub>2</sub>が、このときの復号処理によって得られるはずである。そこで、コードマッチング処理部1002bは、ここで得られた2つのデータがその関係を満たすか否かを診断し、その診断結果を制御処理部1102cに通知する(S1101)。

## 【0050】

制御処理部1102cは、実行可能プログラムファイルM<sub>3</sub>に組み込まれたL<sub>i b</sub>ファイルL<sub>2</sub>が正当使用によるものか否かを、コードマッチング処理部1002bの診断結果に基づき判断する(S1102)。具体的には、1組のプロテクトコード生成に用いた関係を満たさないという診断結果が通知された場合には、プロテクトコードに矛盾ありと判断し、1組のプロテクトコード生成に用いた関係を満たすという診断結果が通知された場合には、プロテクトコードに矛盾なしと判断する。

## 【0051】

以上の自己診断処理によってプロテクトコードに矛盾ありと判断した場合には

、制御処理部1102cは、その旨を表すエラーメッセージを出力し(S1003)、終了処理部1002dに終了指示を与える。この指示に応じて、終了処理部1002dは、所定の終了処理を実行し、プログラムを終了させる(S1106)。

#### 【0052】

一方、プロテクトコードに矛盾なしと判断した場合には、制御処理部1002dは、ゲーム実行処理部1003に開始指示を与える。この指示に応じて、ゲーム実行処理部1003は、ゲーム処理を開始する(S1104)。

#### 【0053】

以後、コントローラを介して送られてくる、ゲームプレイヤからの指示は、入力データ受付処理部1004からゲーム実行処理部1003に渡される。そして、ゲーム実行処理部1003は、入力データ受付処理部1004から渡された指示に応じて、表示装置上の表示映像およびオーディオ装置からの音声を変化させるべく、音声制御処理部1005と映像制御処理部1006とを制御する。この間、ゲーム実行処理部1003は、ゲームオーバーの条件を満たしたか否かの判断(S1105)を定期的に行ない、ゲームオーバーと判断したら、ゲームを終了させる(S1106)。

#### 【0054】

このように、本実施の形態に係るエンタテインメント装置によれば、読み込んだゲーム用実行可能プログラムファイルに正当なプロテクトコードが埋め込まれている場合にだけゲームを開始させ、読み込んだゲーム用実行可能プログラムファイルに正当なプロテクトコードが埋め込まれていなければ、ゲームを終了させる。したがって、フロントエンドツールの正当使用によってゲーム用実行可能プログラムファイルが作成されていなければ、エンタテインメント装置でゲームをすることができない。このため、フロントエンドツールの正当使用者以外のユーザによるL i bファイル使用を止することできる。

#### 【0055】

なお、以上においては、上述の実行可能プログラムファイル作成処理により作成された実行可能プログラムファイルを実行するための装置としてエンタテイン

メント装置を例に挙げたが、上述の実行可能プログラムファイル作成処理により作成された実行可能プログラムファイルを実行するための装置は、必ずしもエンタインメント装置である必要はない。例えば、プログラムの実行能力のある、一般的な電子計算機であってもよい。もちろん、実行可能プログラムファイルも、ゲーム以外の処理を実行させるものであってもよい。

#### 【0056】

また、ここでは、記憶媒体から実行可能プログラムファイルが実行装置に読み込まれるようにしているが、ネットワーク等の伝送媒体によって実行可能プログラムファイルが伝送され、それが実行装置のハードディスクに格納されるようにしてもよい。

#### 【0057】

また、以上においては、ライブラリファイルをプロテクト対象としているが、その他のファイル(例えば、画像データ等が格納されたファイル等の、実行可能ファイル以外のファイル)をプロテクト対象としてもよい。

#### 【0058】

##### 【発明の効果】

本発明によれば、ライブラリ等、実行可能ファイル以外のファイルの不正コピーを防止することができる。

##### 【図面の簡単な説明】

##### 【図1】

本発明の実施の一形態に係る情報処理システムのハードウェア構成図である。

##### 【図2】

本発明の実施の一形態に係る情報処理システムの機能構成図である。

##### 【図3】

本発明の実施の一形態に係る実行可能プログラム作成処理のフローチャートである。

##### 【図4】

本発明の実施の一形態に係る1組のプロテクトコードの生成方法を概念的に示した図である。

## 【図5】

本発明の実施の一形態に係る実施可能プログラムファイルが作成されるまでの処理を概念的に示した図である。

## 【図6】

本発明の実施の一形態に係る実施可能プログラムファイルのデータ構造を概念的に示した図である。

## 【図7】

本発明の実施の一形態に係るエンタテインメント装置およびその周辺機器の外観図である。

## 【図8】

本発明の実施の一形態に係るエンタテインメント装置のハードウェア構成図である。

## 【図9】

本発明の実施の一形態に係る光ディスクの格納データのデータ構造を概念的に示した図である。

## 【図10】

本発明の実施の一形態に係るエンタテインメント装置の機能構成図である。

## 【図11】

本発明の実施の一形態に係る実行可能プログラム作成処理のフローチャートである。

## 【符号の説明】

$L_1$ …暗号化ライブラリファイル	$L_2$ …ライブラリファイル
$M_1$ …ソースファイル	$M_2$ …オブジェクトファイル
$M_3$ …実行可能ファイル	
100…情報処理装置	101…CPU
102…グラフィックス機構	103…主記憶
104…メモリバス制御チップ	105, 106, 109…バス
107…ハードディスクドライブ	108…ハードディスク
110…ブリッジ回路	111…入力装置コントローラ

112…CD-ROMドライブ

120…入力装置

200…入力受付部

202…開発成果記憶部

204…ライブラリ記憶部

206…一時ファイル記憶部

208…一時ファイル削除処理部

210…制御処理部

113…出力装置コントローラ

130…出力装置

201…起動時チェック処理部

203…コンパイル処理部

205…復号化処理部

207…リンク処理部

209…コード付与処理部

700…エンタテインメント装置

801…グラフィックプロセッサ

803…光ディスク制御部

805…サウンドバッファ

807…IOPメモリ

809…バス

800…メインCPU

802…I/Oプロセッサ

804…サウンド再生処理プロセッサ

806…OS-ROM

808…メインメモリ

1000…第一プロテクトコード格納部

1001…第二プロテクトコード格納部

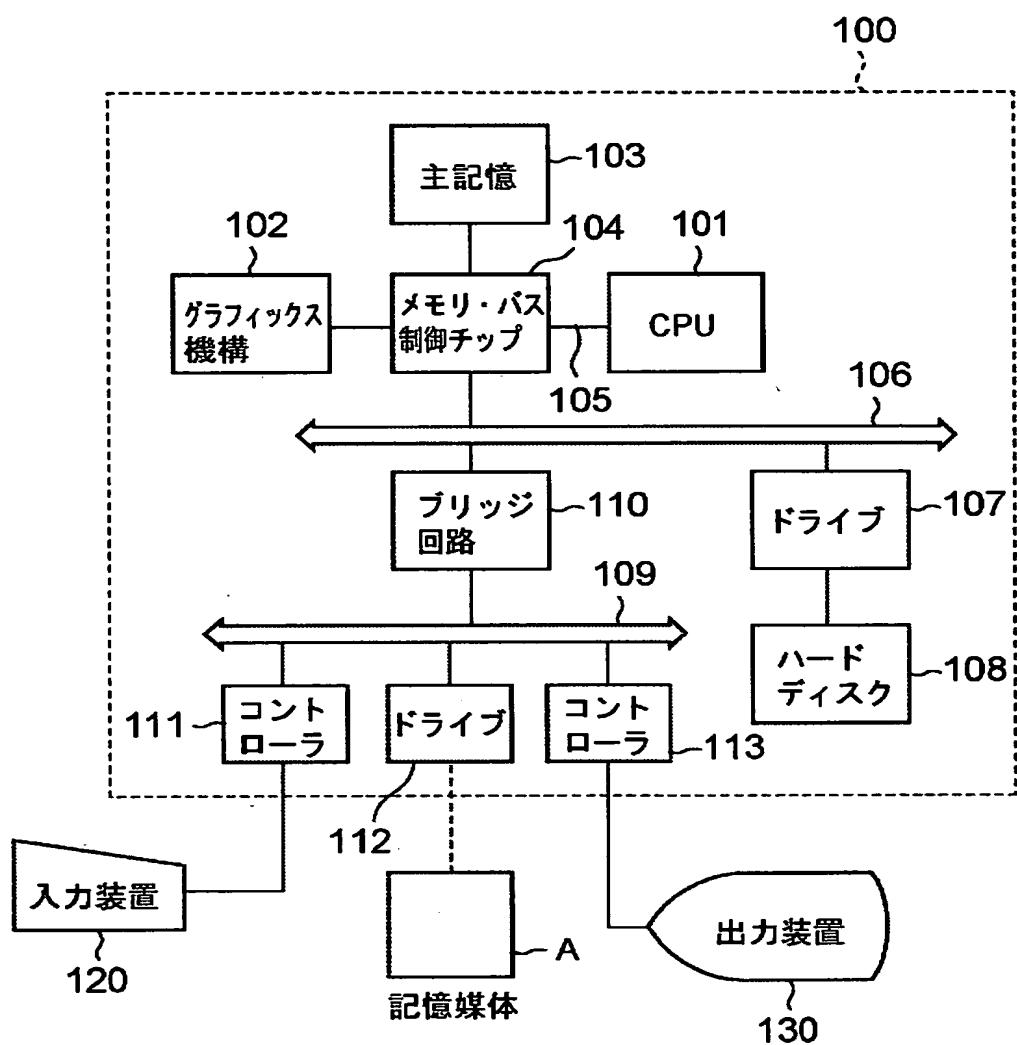
1002…プロテクトコードチェック処理部

1003…ゲーム実行処理部

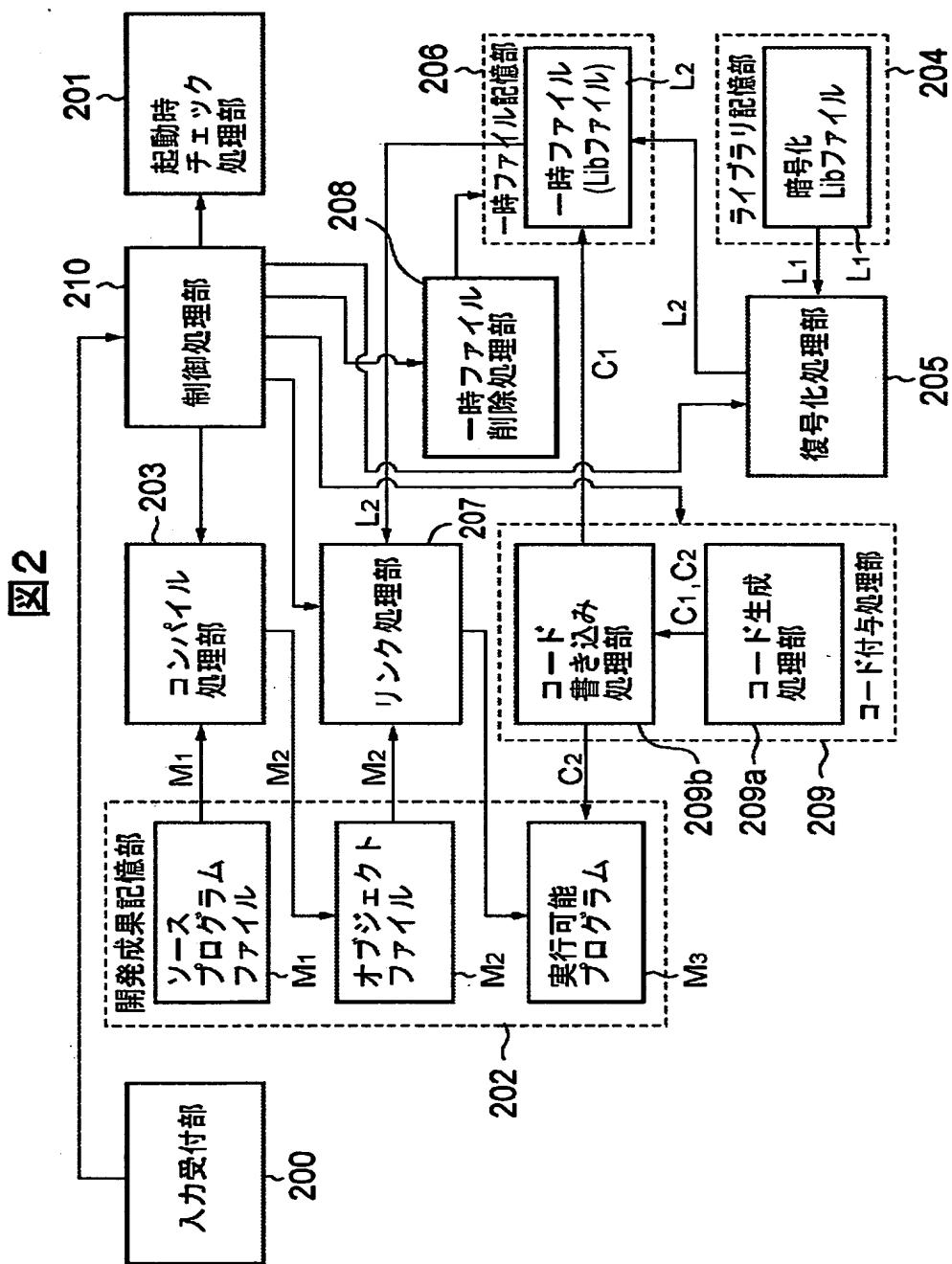
【書類名】図面

【図1】

図1

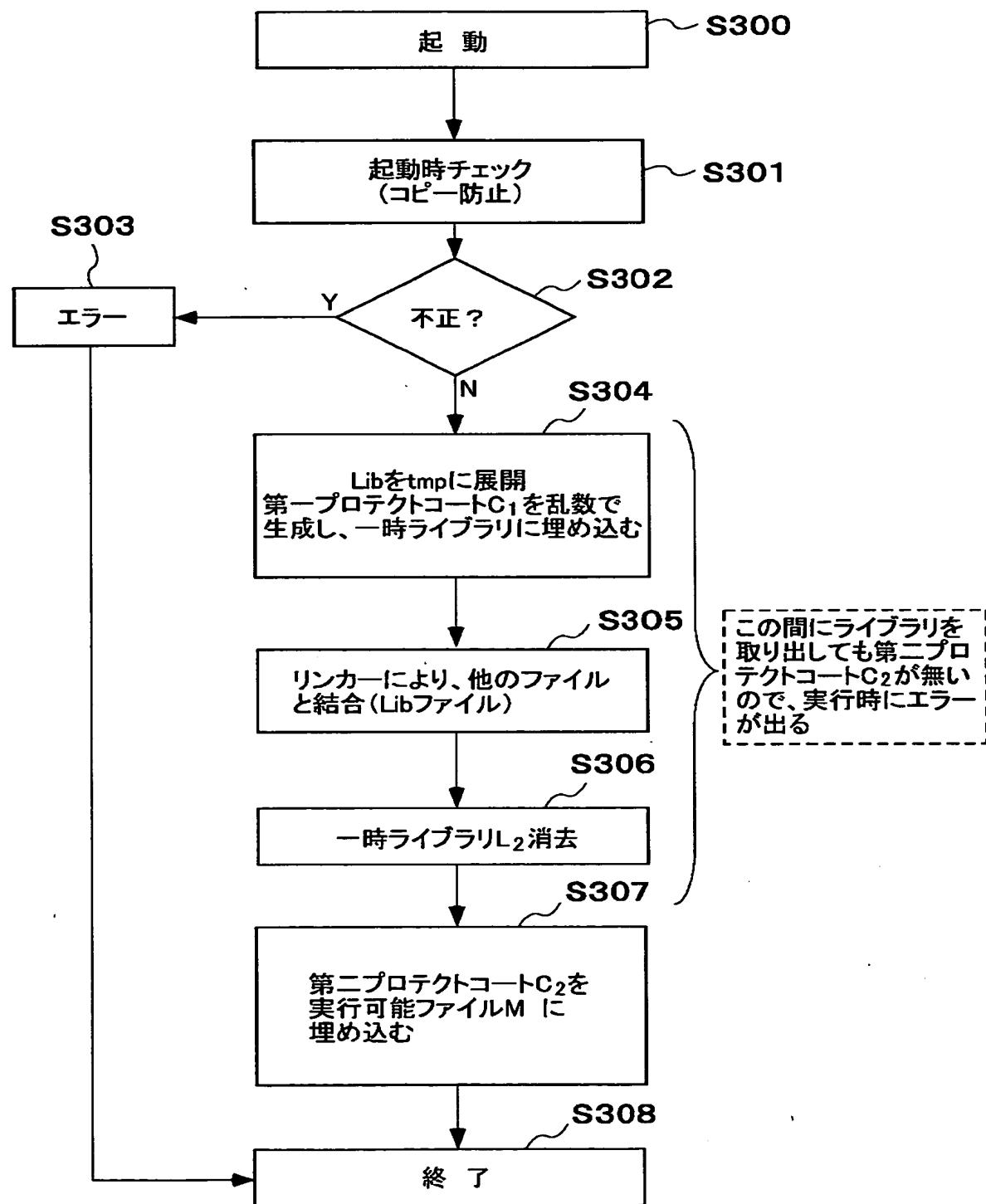


【図2】



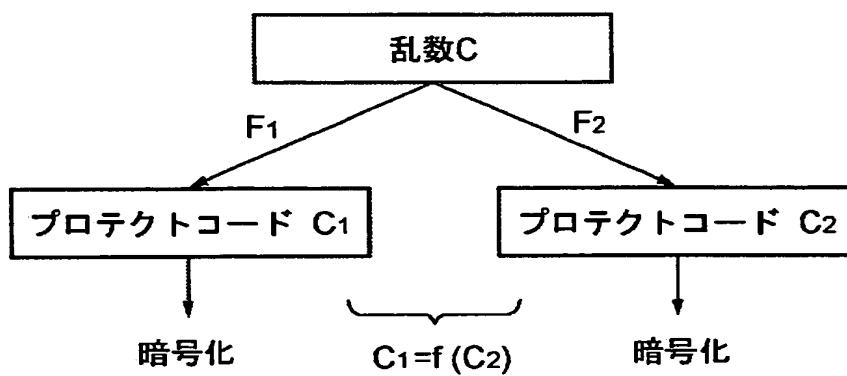
【図3】

図3



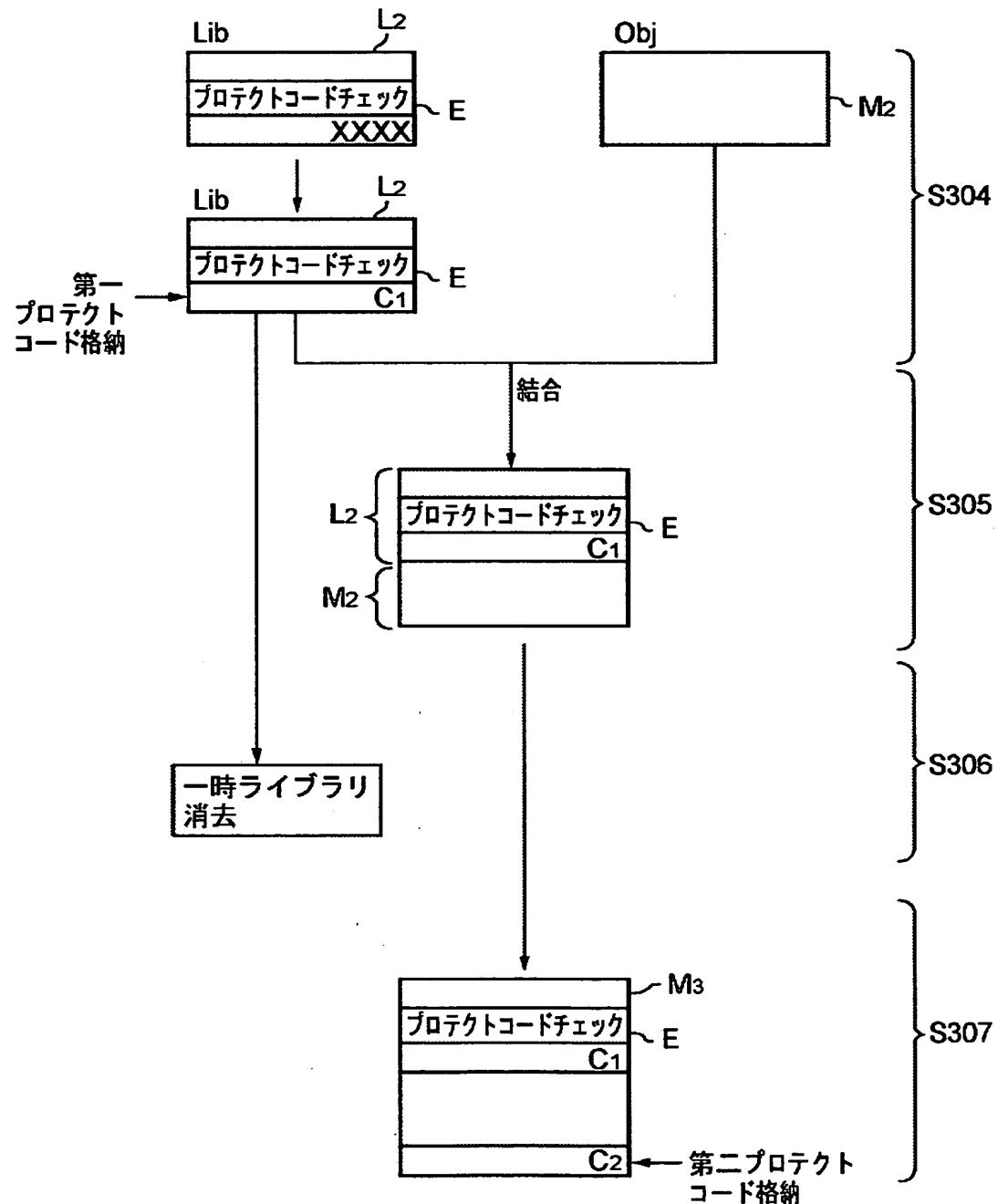
【図4】

図4



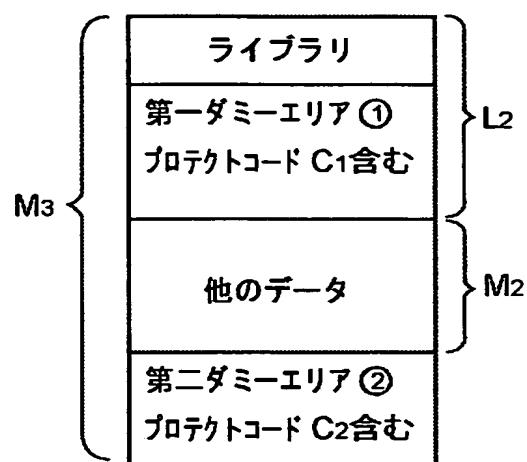
【図5】

図5



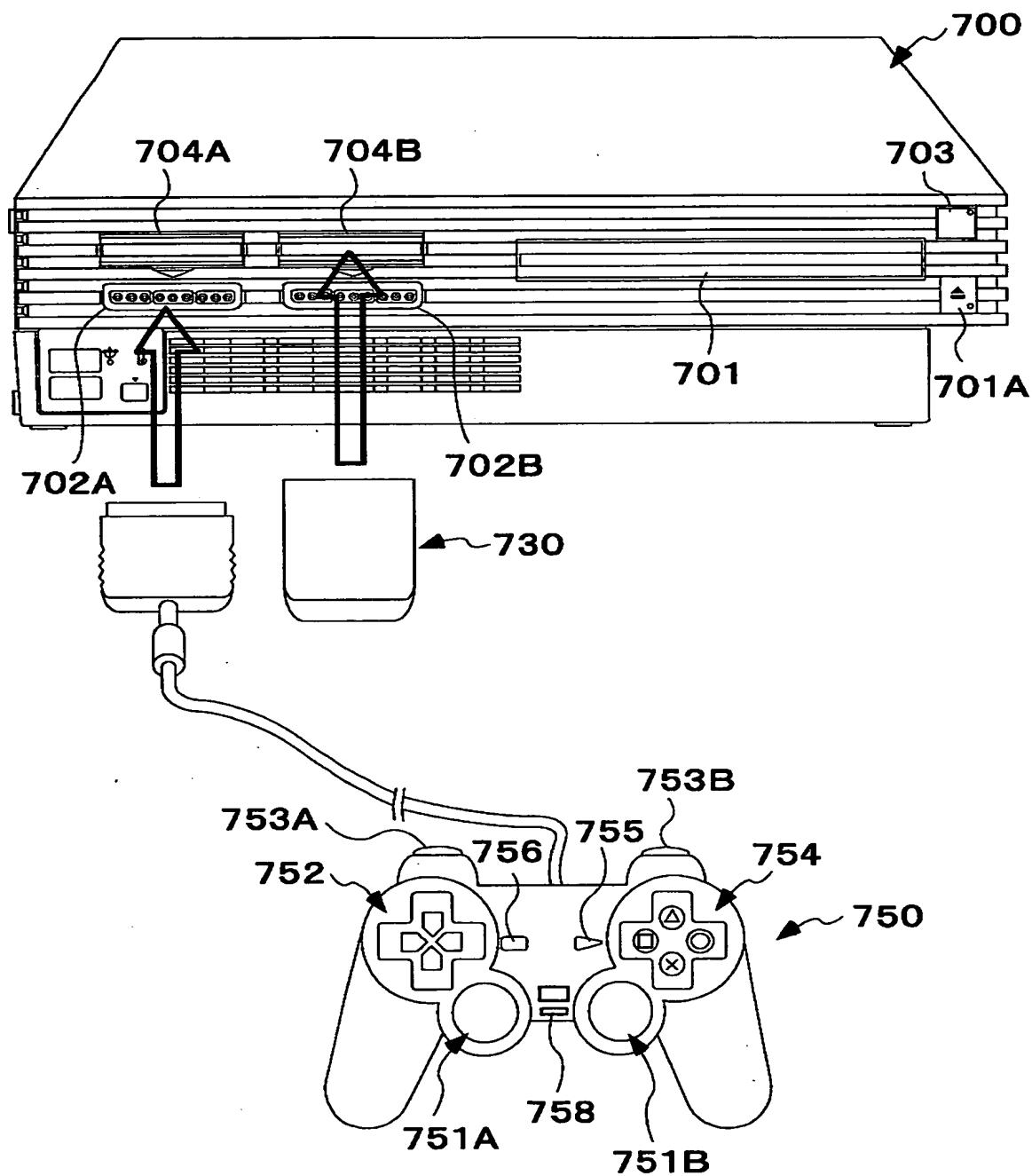
【図6】

図6

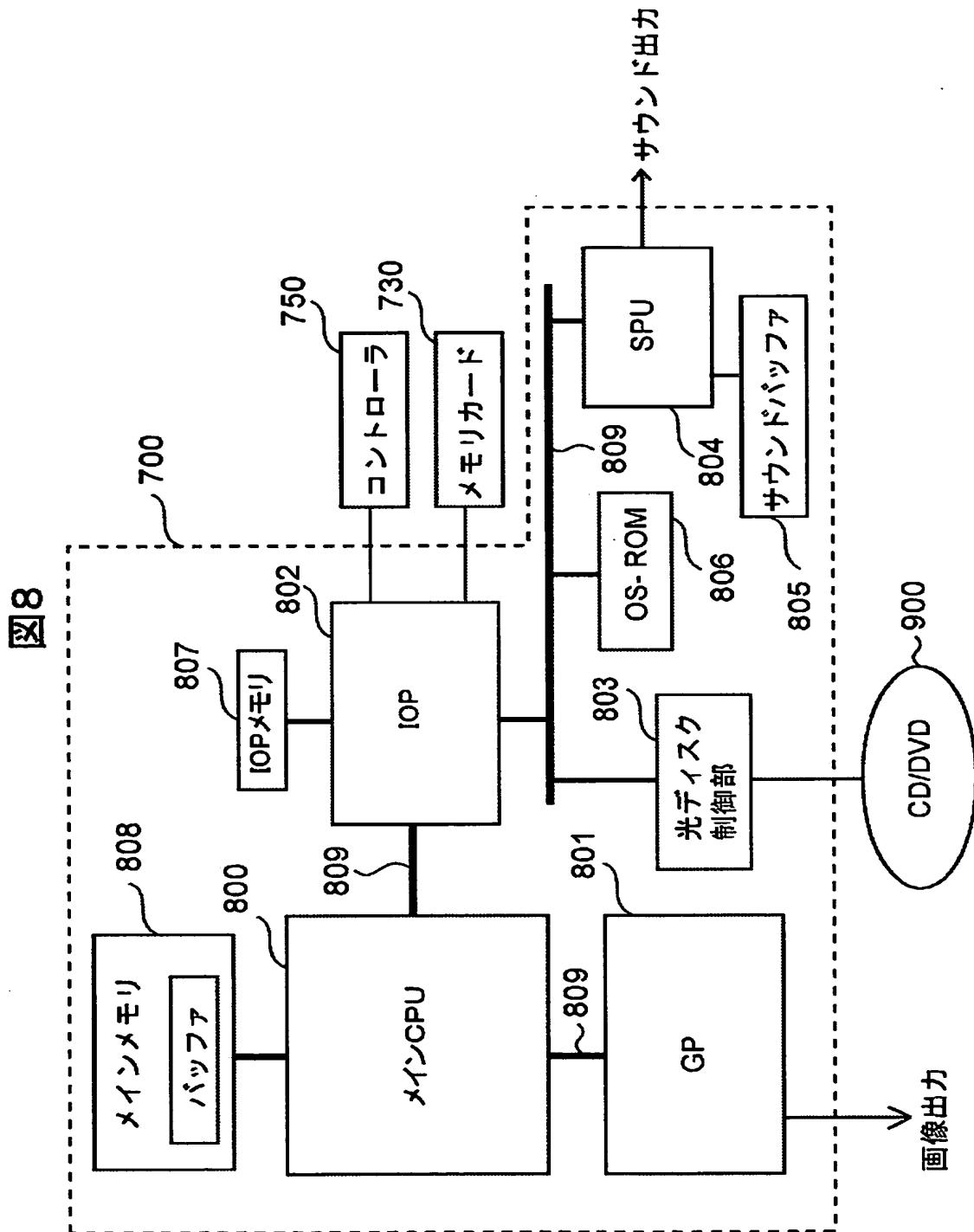


【図7】

図7

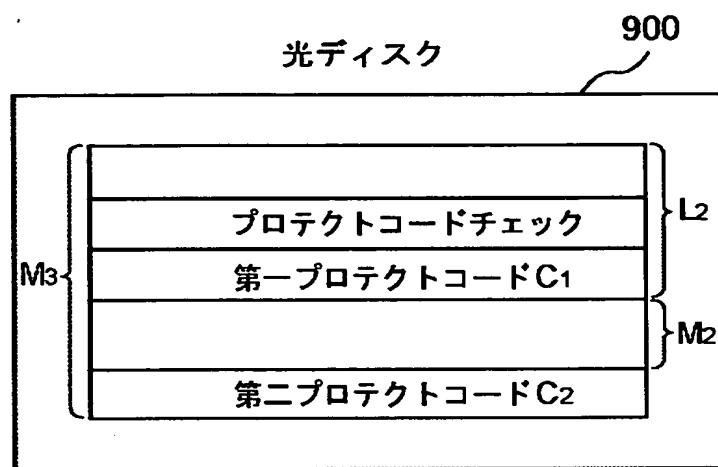


〔図8〕



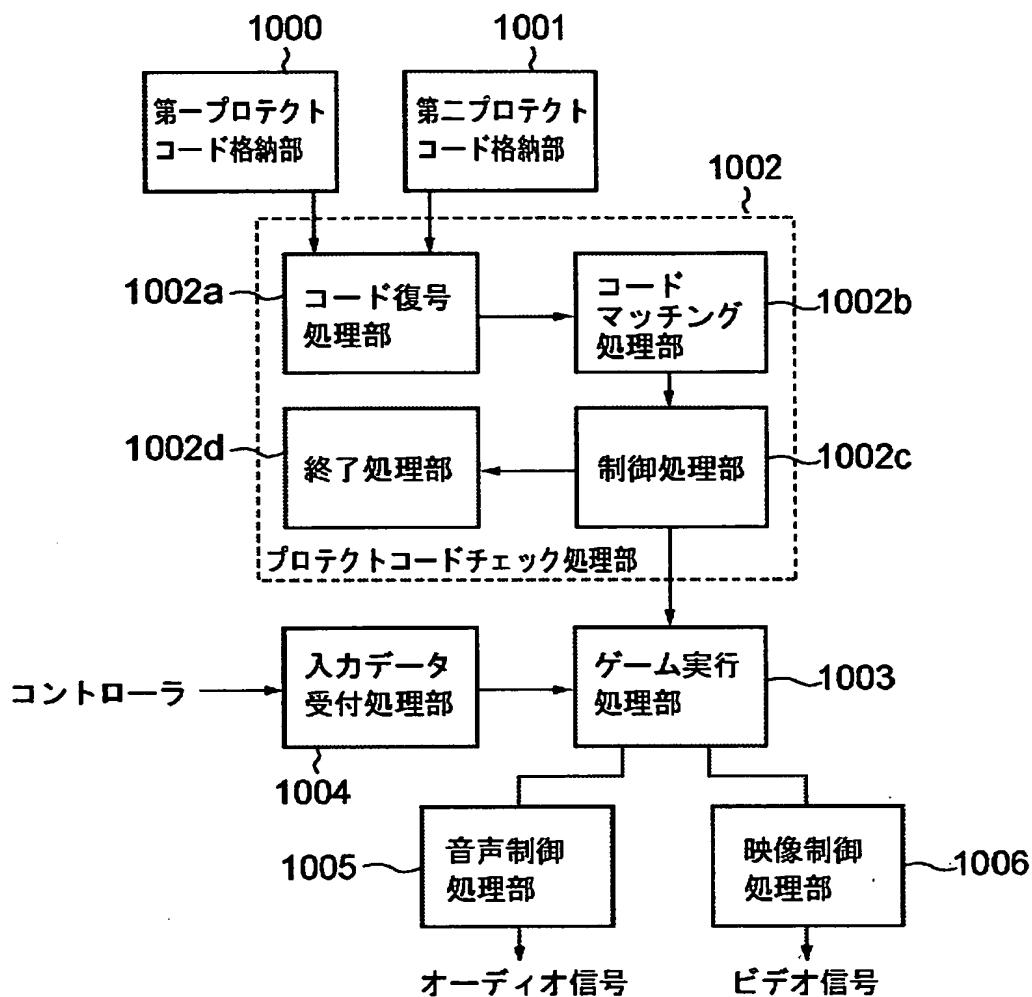
【図9】

図9



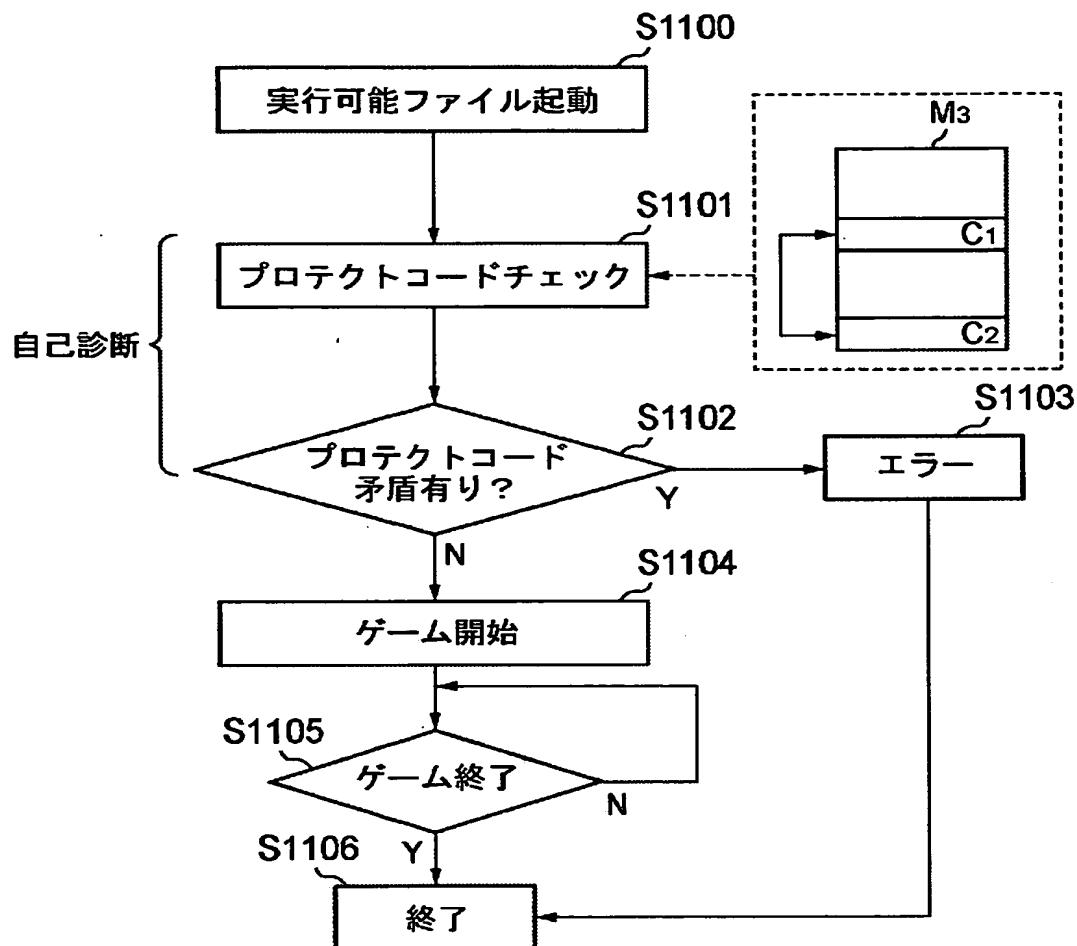
【図10】

図10



【図11】

図11



【書類名】要約書

【要約】

【課題】ライブラリ等の不正コピーを防止する。

【解決手段】暗号化ライブラリファイル $L_2$ には、実行可能プログラムファイルに埋め込まれた2つのプロテクトコードに矛盾があれば、初期化段階で処理を終了させるプロテクトコードチェック手続きが含まれている。実行可能プログラムファイル作成時、暗号化ライブラリファイル $L_2$ が復号され、それに第一プロテクトコード $C_1$ が埋め込まれる(S304)。その後、復号後のライブラリファイル $L_2$ とオブジェクトファイル $M_2$ とが結合編集され、実行可能プログラムファイルが作成される(S305)。そして、使用済みライブラリファイル $L_2$ を消去してから(S306)、実行可能プログラムファイル $M_3$ に、第一プロテクトコード $C_1$ と所定の関係にある第二プロテクトコード $C_2$ がさらに埋め込まれる(S307)。

【選択図】図5

出願人履歴情報

識別番号 [395015319]

1. 変更年月日 1997年 3月31日

[変更理由] 住所変更

住 所 東京都港区赤坂7-1-1

氏 名 株式会社ソニー・コンピュータエンタテインメント